# Unit Testing Exercise: Strings and Regular Expressions

Tasks for exercise in class and for homework to the course ["Programming Advanced for QA" @ SoftUni](https://softuni.bg/trainings/4257/programming-advanced-for-qa-november-2023)

Test your tasks in the Judge system: <https://judge.softuni.org/Contests/4464>

# Unit Test String Method: Repeat String

Look at the **provided skeleton** and examine the RepeatStrings.cs class that you will test:

A screenshot of a computer

Description automatically generated

The method takes in an **array of strings** , and for every string **it repeats it length of the word times** for example the word "**hello**" would be repeated **5 times** because it has **5 letters**:

A computer screen shot of text

Description automatically generated

Then, look at the tests inside the RepeatStringsTests.cs class:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The first test if **finished** so you have a **reference**, the rest of the tests are **empty,** and your task is to finish them. The tests should run when you're finished:

A close up of words

Description automatically generated

# Unit Test String Method: Substring

Test a given method which takes in a **string to be removed and a string as text from which the string to be removed**.

The method is found in the Substring.cs file:

A computer screen shot of a program code

Description automatically generated

You are given a **test** **file** SubstringTests.cs which contains **4 tests**. **One** of them has been **finished partially**, and **three** are **empty** for you to finish:

A screenshot of a computer

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer

Description automatically generated

# Unit Test String Method: Text Filter

Test a given method which takes in an **array of** **strings representing banned words and a string representing text** and **blurs out** every **banned word** foundby **replacing it** with **asterisks**.

The method is found in the TextFilter.cs file:

A computer code on a white background

Description automatically generated

You are given a **test** **file** TextFilterTests.cs which contains **4 tests**. **One** of them has been **finished partially**, and **three** are **empty** for you to finish:

A screenshot of a computer

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer screen

Description automatically generated

# Unit Test String Method: Reverse and Concatenate

Test a given method which takes in an **array of** **strings** putsthe **words** in a **reverse order** and **concatenates them together**.

The method is found in the ReverseConcatenate.cs file:

A computer screen shot of text

Description automatically generated

You are given a **test** **file** ReverseConcatenateTests.cs which contains **6 tests**. **Two** of them has been **finished partially**, and **four** are **empty** for you to finish:

A screenshot of a computer

Description automatically generated

When you are ready make sure your **tests run:**

A close up of a computer screen

Description automatically generated

# Unit Test String Method: Pattern

Test a given method which takes in a **string and a number indicating repetition count** then turning every **even letter** to **lowercase** and every **odd letter** to **uppercase** and **repeats** this process as much times as **specified**.

The method is found in the Pattern.cs file:

A screenshot of a computer code

Description automatically generated

You are given a **test** **file** PatternTests.cs which contains **6 tests**. **One** of them has been **finished partially**, and **five** are **empty** for you to finish:

A screenshot of a computer program

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer screen

Description automatically generated

# Unit Test Regular Expression: Match Names

Test a given method which takes in a **string array of names** and matches names in the form of Firstname Lastname starting both with **capital letters**.

The method is found in the MatchNames.cs file:

A screen shot of a computer code

Description automatically generated

You are given a **test** **file** MatchNamesTests.cs which contains **3 tests**. **One** of them has been **finished**, and **two** are **empty** for you to finish:

A screenshot of a computer program

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer

Description automatically generated

# Unit Test Regular Expression: Match Phone Numbers

Test a given method which takes in a **string array of phone numbers** and matches phones in the form of +359 followed by **either** a **space** or a **hyphen**, then the **area code '2,**' followed by **three** **digits**, and finally, **four** **more** **digits** at the end.

The method is found in the MatchPhoneNumbers.cs file:

A computer code with many colored text

Description automatically generated with medium confidence

You are given a **test** **file** MatchPhoneNumbersTests.cs which contains **4 tests**. **One** of them has been **finished partially**, and **three** are **empty** for you to finish:

A screenshot of a computer

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer

Description automatically generated

# Unit Test Regular Expression: Match Dates

Test a given method which takes in a **string array of dates** and matches dates in the form of **two** **digits** for the **day**, **separated** by a **dash**, **period**, **forward** **slash**, or **backslash**, followed by the **abbreviated** **month** **name** and another occurrence of the same **separator**, ending with a **four-digit year**.

The method is found in the MatchDates.cs file:

A computer screen shot of text

Description automatically generated

You are given a **test** **file** MatchDatesTests.cs which contains **5 tests**. **Two** of them has been **finished partially**, and **three** are **empty** for you to finish:

A screenshot of a computer program

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer code

Description automatically generated

# Unit Test Regular Expression: Match URLs

Test a given method which takes in a **string array of URLs** and matches them in the standard HTTP or HTTPS **format**, **optionally** preceded by **'**www.', and consisting of **valid** **characters** for **domain** **names** and **query** **parameters**.

The method is found in the MatchUrls.cs file:

A screen shot of a computer code

Description automatically generated

You are given a **test** **file** MatchUrlsTests.cs which contains **5 tests**. **Two** of them has been **finished partially**, and **three** are **empty** for you to finish:

A screenshot of a computer

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer

Description automatically generated

# Unit Test Regular Expression: Email Validator

Test a given method which takes in a **string array of emails** and matches **valid emails** which are **combination** of **letters**, **numbers**, **dots**, **underscores**, **percentage** **signs**, **plus** **signs**, or **hyphens** before the '**@**' **symbol**, followed by a **domain** containing **letters**, **numbers**, **hyphens**, and **dots**, and **ending** with a **top-level domain** of at least **two** **letters**.

The method is found in the EmailValidator.cs file:

A computer code with text

Description automatically generated with medium confidence

You are given a **test** **file** EmailValidatorTests.cs which contains **6 test cases for you to write**:

A screenshot of a computer program

Description automatically generated

When you are ready make sure your **tests run:**

A screenshot of a computer screen

Description automatically generated

Finally make sure **all** tests run:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated